Abstract Presentation Back to Agenda
Next Abstract

Custom Reduction of Arithmetic in Linear DSP Transforms

Smarahara Misra James C. Hoe Markus Püschel*
Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

Introduction

In multiplier-less hardware implementations of DSP transforms, multiplication-by-constants are implemented as a network of (wire-)shifts and additions. The number of additions required can be reduced by approximating the multiplicative constants using lower precision fixed-point representations, but the loss of precision increases the numerical error in the implementation. This trade-off can be leveraged to reduce the hardware area, critical path and power/energy while maintaining the perceptible quality of a signal processing application (e.g., MPEG-4). This paper describes an automatic approach to minimize the number of additions subject to a given quality measure, or, vice-versa, to maximize the quality subject to a given number of available additions. Our automatic approach can handle linear DSP transforms in general and includes optimizations over the space of algorithm design. A Verilog backend generates synthesizable descriptions of the final variable-width fixed-point implementations.

Approach

We consider the following two optimization problems for a given linear DSP transform: (1) Given a quality threshold Q, find the multiplierless implementation with the least arithmetic cost C that satisfies Q; (2) Given an arithmetic cost threshold C, find the multiplierless implementation with the highest quality Q. Our proposed system automatically solves this problem in the following steps. We consider problem (1); problem (2) is analogous.

Given is a formally specified linear DSP transform T (e.g., a DCT of size 8) and the quality threshold Q (e.g., the maximum allowed error of the output).

Step 1: Generating a Fast Algorithm. First, we generate a fast algorithm for T represented as a formula in a mathematical notation using SPIRAL[1]. The formula is built from few constructs and primitives such as the Kronecker product '$\otimes$', permutation matrices, or 2×2 rotations R_α. For example, one out of many possible formulas for the DCT of size 8 looks like

$$\begin{aligned}\mathrm{DCT}_8 \;=\; & [(2,5)(4,7)(6,8),8] \cdot (\mathrm{diag}(1, \tfrac{1}{\sqrt{2}}) \oplus \mathrm{R}_{\frac{3}{8}\pi} \oplus \mathrm{R}_{\frac{15}{16}\pi} \oplus \mathrm{R}_{\frac{21}{16}\pi}) \\ & \cdot [(2,4,7,3,8),8] \cdot ((\mathrm{F}_2 \otimes \mathbf{1}_3) \oplus \mathbf{1}_2) \cdot (\mathbf{1}_4 \oplus \mathrm{R}_{\frac{3}{4}\pi} \oplus \mathbf{1}_2) \cdot [(2,3,4,5,8,6,7),8] \\ & \cdot (\mathbf{1}_2 \otimes ((\mathrm{F}_2 \oplus \mathbf{1}_2) \cdot [(2,3),4] \cdot (\mathbf{1}_2 \otimes \mathrm{F}_2))) \cdot [(1,8,6,2)(3,4,5,7),8].\end{aligned}$$

Step 2: Manipulation for Numerical Stability. In the second step, we formally manipulate the formula to increase its numerical stability, which determines how quick the quality of T degrades when implemented in low precision. In particular, we expand the formula into lifting steps using ideas from[2].

*The work of Markus Püschel was supported by DARPA through research grant DABT63-98-1-0004 administered by the Army Directorate of Contracting and by NSF through award 9988296

[1] J. Moura, J. Johnson, R. Johnson, D. Padua, V. Prasanna, M. Püschel, B. Singer, M. Veloso, and J. Xiong. Generating Platform-Adapted DSP Libraries using SPIRAL. In *Proc. HPEC*, 2001. `http://www.ece.cmu.edu/~spiral`.

[2] J. Liang and T.D. Tran. Fast Multiplierless Approximations of the DCT With the Lifting Scheme. In *IEEE Transactions on Signal Processing*, Vol.49, No.12, Dec 2001, pages 3032-3044.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **20 AUG 2004**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Custom Reduction of Arithmetic in Linear DSP Transforms**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

See also ADM001694, HPEC-6-Vol 1 ESC-TR-2003-081; High Performance Embedded Computing (HPEC) Workshop (7th)., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**30**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

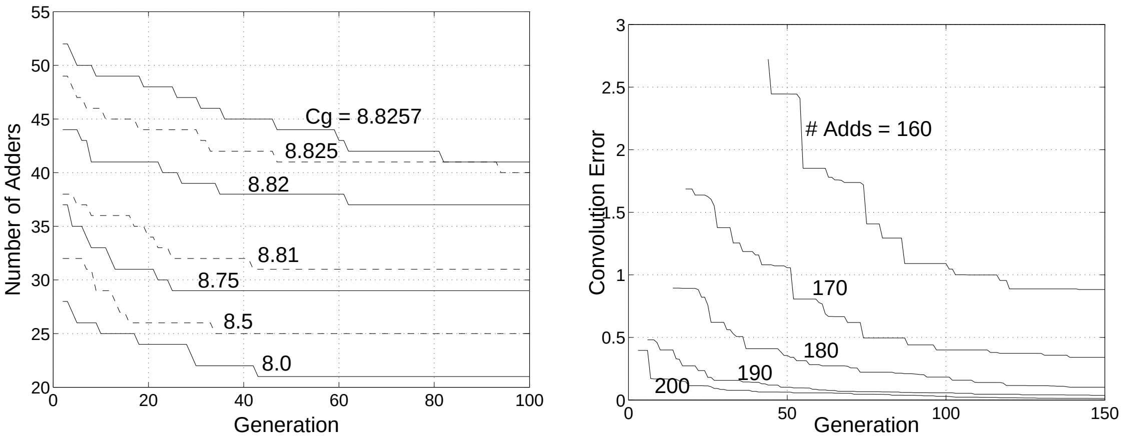


Figure 1: Evolutionary optimization. Left: for DCT_8 minimizing number of additions for various given coding gains (cg); right: for DFT_{16} optimizing convolution error for various given numbers of additions.

Step 3: Constant Reduction and Search. In this step the actual constrained optimization is performed using an automated search. The idea is to replace each occurring constant (multiplication) in the formula by a low-precision version, specified by the number of bits in $\{0, 1, \ldots b_{max}\}$. Doing so for every constant yields an approximation $\tilde{T}$ of the original transform T; $\tilde{T}$ has a lower cost C than the original, i.e., requires less additions. If the formula for T contains n multiplier constants $a_1, \ldots, a_n$, there are $(b_{max} + 1)^n$ many ways of approximation, which determines the search space for our optimization. Since an exhaustive search is infeasible we use evolutionary and greedy search techniques to find the approximation with the lowest cost (least number of additions) that still satisfies the quality threshold Q.

Step 4: Mapping to Verilog. In this final step we map the found (approximated) formula into Verilog.

We note that in the above the approach, the formula, i.e., algorithm chosen for the transform was fixed. The optimization can readily be extended to include the space of different possible formulas into the optimization using SPIRAL's formula generator.

Experimental Results

We show two examples for two different optimization problems for the discrete cosine transform (DCT) and discrete Fourier transform (DFT).

DCT, size 8. We chose as quality measure coding gain (cg) in dB, which for the exact (infinite precision) DCT is about 8.8259. We considered one formula for the DCT generated by SPIRAL (similar to the one above). A 10-bit multiplierless implementation for this formula requires 56 additions. After formula manipulation, we considered 9 constants in the formula for further approximation, which yields a search space of size 9^{10}. Figure 1 (left) shows the results of an evolutionary search for various cg thresholds. The abscissa shows the generations in this search, the ordinate the found solution with the least cost. For example, after 100 generations, for cg = 8.81 a solution with only 31 adders was found. The search took 30 minutes.

DFT, size 16. We chose as quality measure the convolution error (ce), which determines to what extent the DFT's convolution property is violated. The exact DFT has ce = 0. Again we considered one particular formula, whose 10-bit implementation requires 256 adders. Figure 1 (right) shows the results for fixing the number of additions and optimizing the achievable quality. For example, by allowing 170 adders, a solution with ce = 0.341 was found after 150 generations. The search took 2 hours.

Custom Reduction of Arithmetic in Linear DSP Transforms

S. Misra, A. Zelinski, J. C. Hoe, and M. Püschel
Dept. of Electrical and Computer Engineering
Carnegie Mellon University

Research Overview

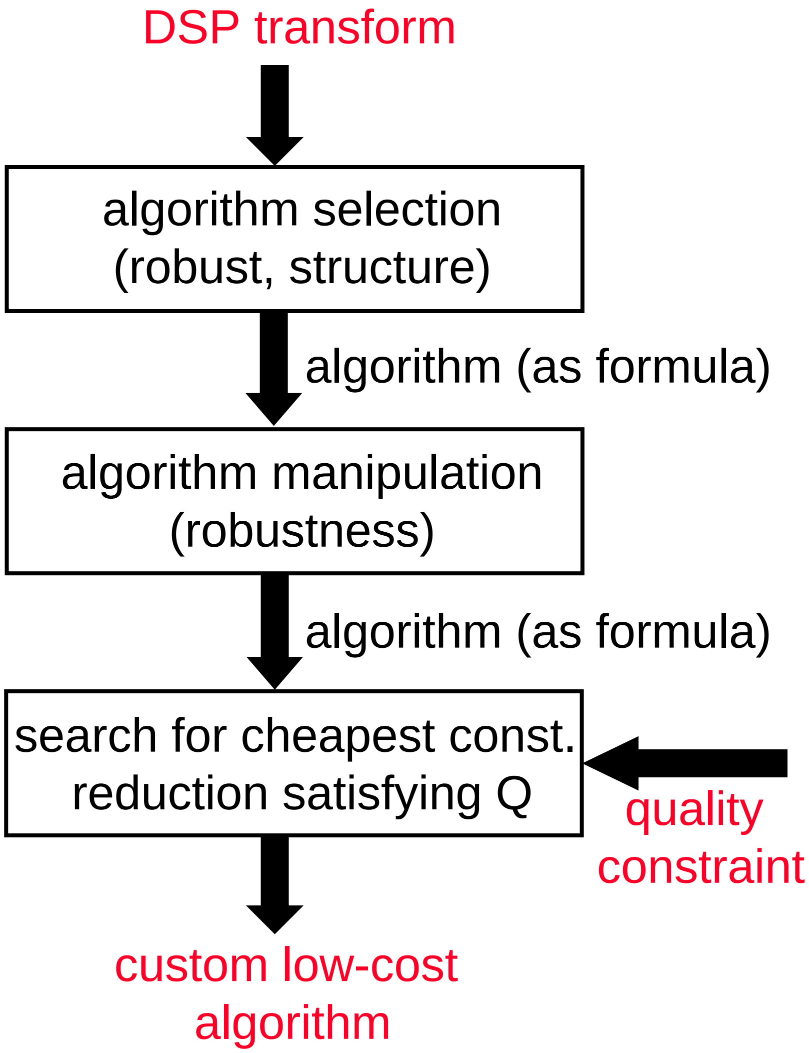
- Linear DSP transforms
  - e.g. DFT, DCTs, WHT, DWTs, ….
  - ubiquitously used, often in computation intensive kernels
  - comprised of additions and multiplication-by-constant
  - applications: multimedia, bio-metric, image/data processing . . . .
- Light-weight hardware implementations
  - fixed-point data format
  - multiplierless: mult-by-constant as shifts and adds
  - problem 1: output quality reduced by cost-saving measures

    (reducing the bitwidth of data and constants)
  - problem 2: different applications have vastly different quality metric and requirements

    ⇒ need application specific tuning

Our Goal: automatic, custom reduction of arithmetic (additions) w.r.t. a given application's requirements

Our Automatic Flow



DSP transform

algorithm selection (robust, structure)

algorithm (as formula)

algorithm manipulation (robustness)

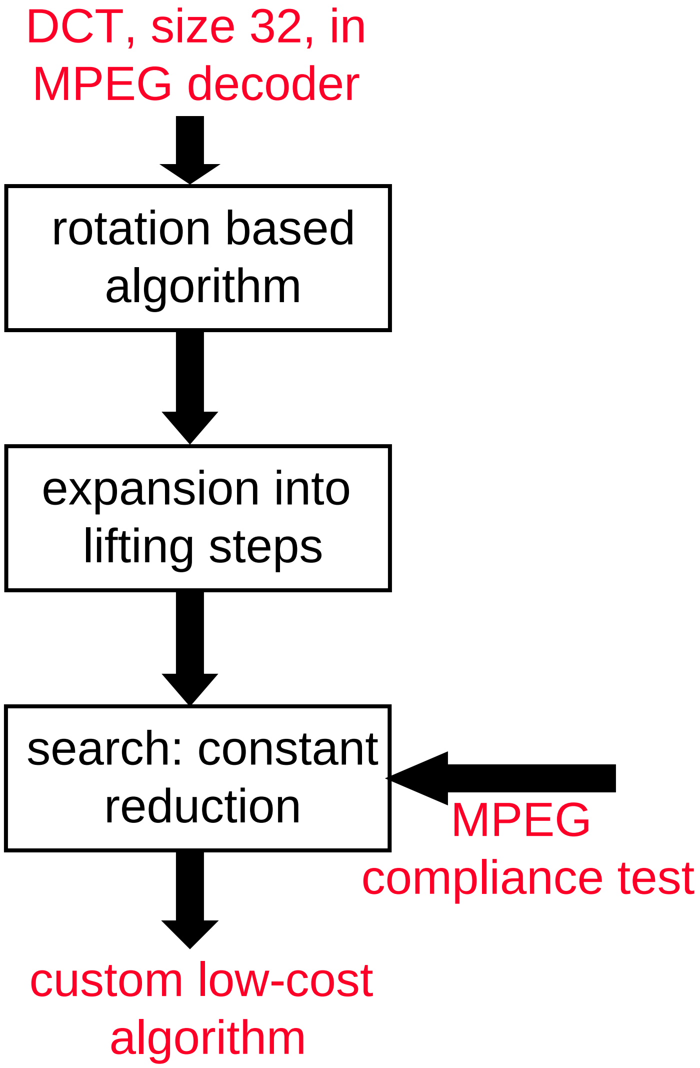
algorithm (as formula)

search for cheapest const. reduction satisfying Q

quality constraint

custom low-cost algorithm

Example



DCT, size 32, in MPEG decoder

rotation based algorithm

expansion into lifting steps

search: constant reduction

MPEG compliance test

custom low-cost algorithm

Related Work

- Liang/Tran, “Fast Multiplierless Approximation of the DCT with the Lifting Scheme,” IEEE Trans. Sig. Proc., 49(12) 2001, pp. 3032-3044
  - examined arithmetic cost reduction for DCT size 8
  - steps performed by hand, exhaustive search

- Fang/Rutenbar/Püschel/Chen, “Toward Efficient Static Analysis of Finite-Precision Effects in DSP Applications via Affine Arithmetic Modeling,” Proc. DAC 2003
  - efficient static analysis of output error (hard and probabilistic)
  - range of input values used/needed
  - analysis assumes a common global bitwidth

- Püschel/Singer/Voronenko/Xiong/Moura/Johnson/Veloso/Johnson, “SPIRAL system”, www.spiral.net
  - automatic generation of custom runtime optimized DSP transform software
  - provides implementation environment for our approach (in particular algorithm generation and manipulation)

Outline

- DSP transform algorithms
- Algorithm manipulation for robustness
- Multiplication by constants
- Search Methods
- Results

DSP Algorithms as Formulas: Example DFT size 4

Cooley/Tukey FFT (size 4):

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fourier transform **Diagonal matrix (twiddles)**

$$DFT_4 = (DFT_2 \otimes I_2) \cdot diag(1,1,1,i) \cdot (I_2 \otimes DFT_2) \cdot [(2,3),4]$$

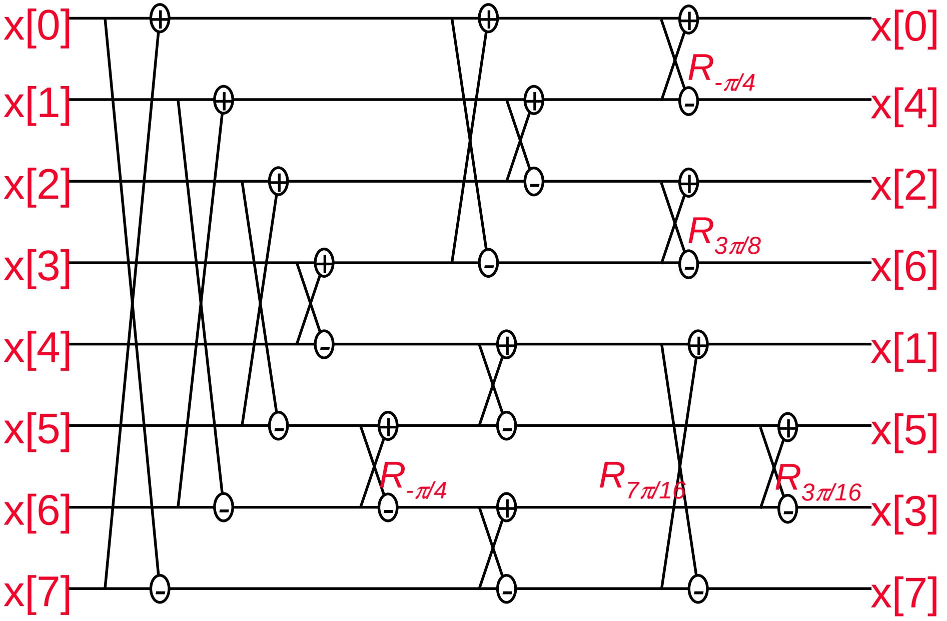
Kronecker product **Identity** **Permutation**

allows for computer generation/manipulation (provided by SPIRAL)

Example: DCT size 8

$$[(2,5)(4,7)(6,8),8]$$
$$\cdot (diag(1,1/\sqrt{2}) \oplus R_{3\pi/8} \oplus R_{15\pi/16} \oplus R_{21\pi/16})$$
$$\cdot [(2,4,7,3,8),8] \cdot ((DFT_2 \otimes I_3) \oplus I_2) \cdot [(5,6),8]$$
$$\cdot (I_4 \oplus 1/\sqrt{2} \cdot DFT_2 \oplus I_2) \cdot [(2,3,4,5,8,6,7),8]$$
$$\cdot (I_2 \otimes ((DFT_2 \oplus I_2) \cdot [(2,3),4] \cdot (I_2 \otimes DFT_2)))$$
$$\cdot [(1,8,6,2)(3,4,5,7),8]$$

as formula
(generated by SPIRAL)



as data flow diagram

Basic building blocks:

- 2 x 2 rotations, DFT_2's (butterflies), permutations, diagonal matrices (scaling)

Algorithm is orthogonal = robust to input errors (from fixed point representation)

Outline

- DSP transform algorithms
- Algorithm manipulation for robustness
- Multiplication by constants
- Search Methods
- Results

Fixed Point Error: Data vs. Transform

Implementing a transform $x \alpha \; Tx$ in fixed point arithmetic produces two type of errors:

- Error in input x: $\| x - \tilde{x} \|$
  - from rounding of the input coefficients *x* to the fix-point data representation $\tilde{x}$
  - for robustness: choose orthogonal algorithms

- Error in transform: $\| T - \tilde{T} \|$
  - from finite precision multiplication by constants

    further approximation is a source of savings in multiplierless implementations
  - for robustness: translate algorithm into lifting steps

Lifting Steps

- Lifting step (LS): $\begin{bmatrix} 1 & x \\ 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & 0 \\ y & 1 \end{bmatrix}$
  - invertible (det = 1) independent of approximation of x, y
  - inverse of LS is also LS *(with –x, -y)*

    ∴ *if LS is cheap, then so is its inverse*
- Rotation as lifting steps

target for approximation

$$R_\alpha = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ u & 1 \end{bmatrix} \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix}$$

$$p = \frac{1-\cos\alpha}{\sin\alpha} = \tan\frac{\alpha}{2}, \qquad u = -\sin\alpha$$

➡ rotation based algorithms can be automatically expanded into LS

Error Analysis

- rounding error in the first lifting step (third LS analogous)

$$\tilde{R}_\alpha = R_\alpha + \begin{bmatrix} 1 & \varepsilon \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ u & 1 \end{bmatrix} \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} = R_\alpha + \begin{bmatrix} -\varepsilon \sin\alpha & \varepsilon \cos\alpha \\ 0 & 0 \end{bmatrix}$$

not magnified

- rounding error in the second lifting step

$$\tilde{R}_\alpha = R_\alpha + \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} = R_\alpha + \begin{bmatrix} \varepsilon \tan\frac{\alpha}{2} & \varepsilon \tan^2\frac{\alpha}{2} \\ \varepsilon & \varepsilon \tan\frac{\alpha}{2} \end{bmatrix}$$

ε is magnified, unless α in [0, π/2] or [3π/2, 2π]

Solution: angle manipulation

$$R_\alpha = R_{\alpha-\pi/2} \cdot R_{\pi/2} = R_{\alpha-\pi/2} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Ensuring Robustness

Steps to ensure robustness

- Choose algorithms based on rotations
- Manipulate angles of rotations
- Expand into lifting steps

➡ *Done automatically as formula manipulation*

Outline

- DSP transform algorithms
- Algorithm manipulation for robustness
- Multiplication by constants
- Search Methods
- Results

Multiplication by Constants

Operations in transforms:

$y = x_1 + x_2$ additions

$y = cx$ multiplication by constant

Example:

simple $c=0.10111011 =$ → 5 adds (5 shifts)

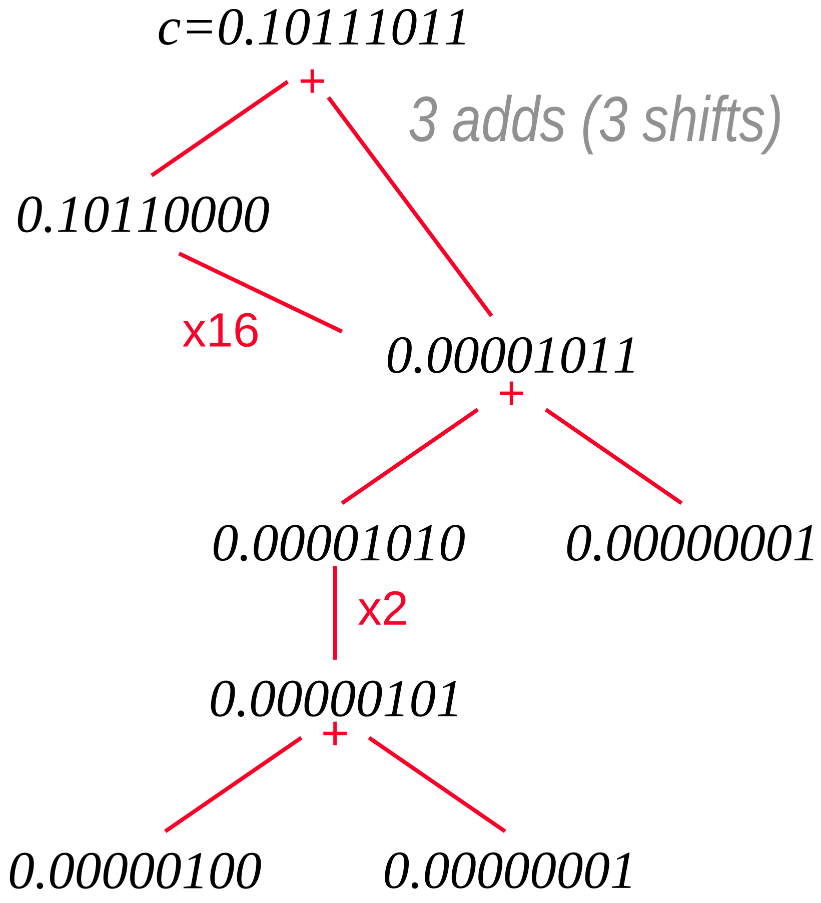
SD recoding 1 $c=0.1100\bar{1}10\bar{1}$ → 4 adds (3 shifts)

SD recoding 2 $c=0.11000\bar{1}0\bar{1}$ → 3 adds (3 shifts)

SD recoding is not optimal

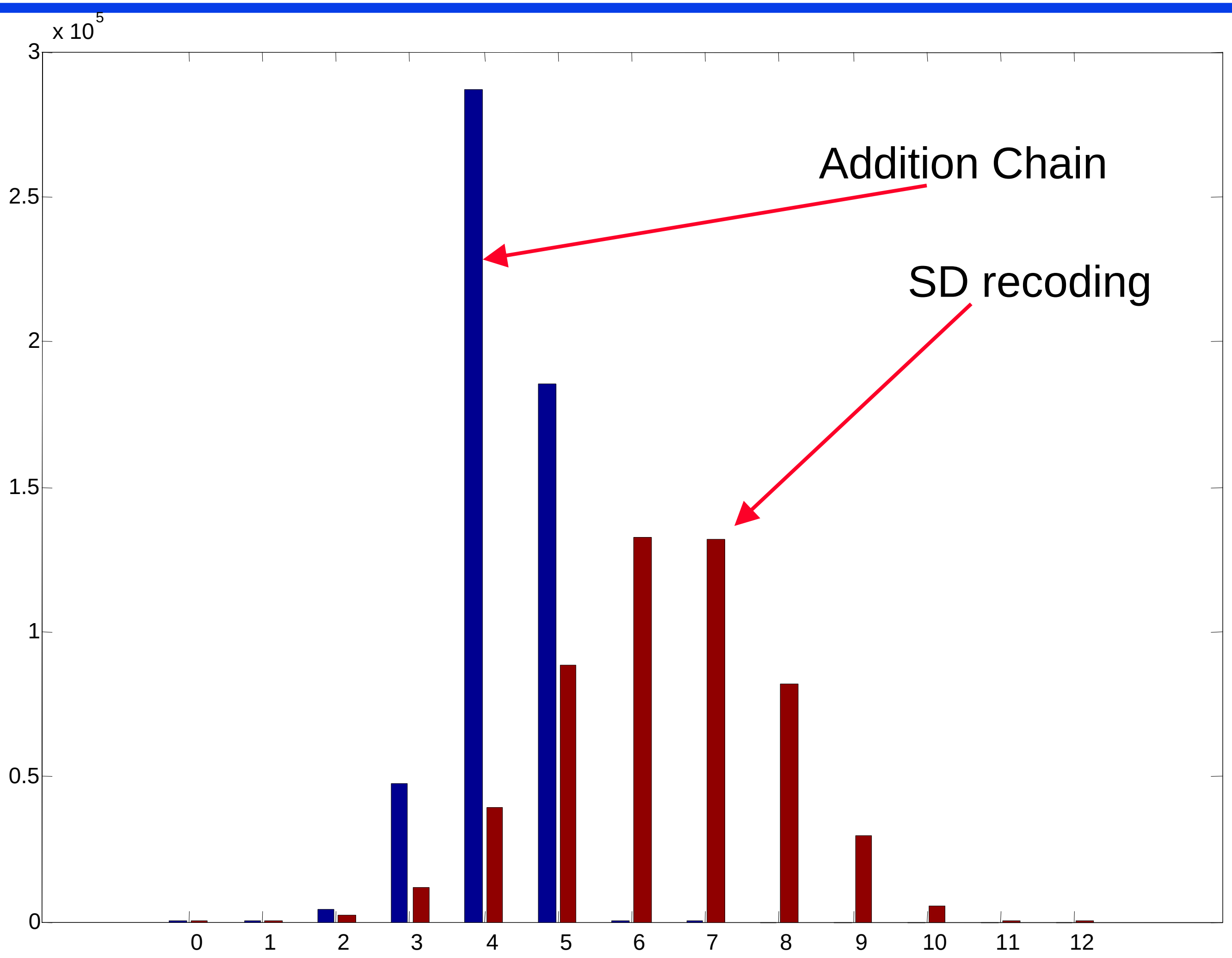
Addition/Subtraction Chain

- Provide optimal solution for constant mult using adds and shifts
- Finding the optimal addition chain is a hard problem
- A near optimal table of solutions can be computed using dynamic programming methods*
- For all constants up to 2^{19}
  - only 225 constants require more than 5 additions *(214@6, 11@7)*



**Sebastian Egner, Philips Research, Eindhoven*

SD recoding vs. Addition Chains



Histogram of addition cost for all constants between 1 and 2^{19}

Outline

- DSP transform algorithms
- Algorithm manipulation for robustness
- Multiplication by constants
- Search Methods
- Results

Optimization Problem

Given a linear DSP transform and quality measure Q

1. Find the multiplierless implementation with the least arithmetic cost C (number of additions) that satisfies a given Q threshold

2. Find the multiplierless implementation with the highest quality Q for a given arithmetic cost C threshold

Quality Measures of Transforms

For an approximation $\tilde{T}$ of a transform T.

- Transform independent Q
  - $\| T - \tilde{T} \|$ for some norm || · ||
- Transform dependent Q
  - coding gain for DCT
  - convolution error for DFT
- Application-based Q
  - MPEG standard compliance test

Search Space: approximating multiplicative constants

- For each multiplication-by-constant in the transform choose custom bitwidth $i \in [0 \ldots k-1]$
  - Given n constants, k^n configurations are possible

- But, for a given constant, not all k configurations lead to different cost,

e.g., given 5-bit constant 0.11101, SD recoding gives

5-bit = .11101	= 1.00$\bar{1}$01	⇒ 2 adds
4-bit = .1110	= 1.00$\bar{1}$0	⇒ 1 adds
~~3-bit = .111~~	~~= 1.00$\bar{1}$~~	~~⇒ 1 adds~~
~~2-bit = .11~~	~~= 0.11~~	~~⇒ 1 adds~~
1-bit = .1	= 0.1	⇒ 0 adds
0-bit = 0	= 0	⇒ 0 adds

Recall all constants up to 19-bits can be reduced to 5 adds

Search Methods

- Global Bitwidth
  - all constant assigned the same bitwidth
  - *very fast (small search space), but only works well in some cases*
- Greedy Search
  - starting with maximum bitwidth, in each round, choose one constant to be reduced by 1-bit that minimizes quality loss

    (also go bottom-up instead of top-down)
  - *local minima traps are possible*
- Evolutionary Search
  - start with a population of random configurations
  - in each round
    1. breed a new generation by crossbreeding and mutations
    2. select from generation the fittest members
    3. repeat new round
  - *local minima traps*

Outline

- DSP transform algorithms
- Algorithm manipulation for robustness
- Multiplication by constants
- Search Methods
- Results

Interaction between Transforms, Q and Search

- Goal: given a transform and a required Q threshold, find an approximation to the transform that requires the fewest additions
- Transforms and Q tested

Transform	*Quality Threshold*
8-pt. DCT-II	8.82 dB coding gain (cg)
16-pt. DFT	Convolution error = 1
32-pt. DCT-II	Limited Compliance (LC) MP3 decoder♣
18x36 IMDCT	LC MP3 decoder♣

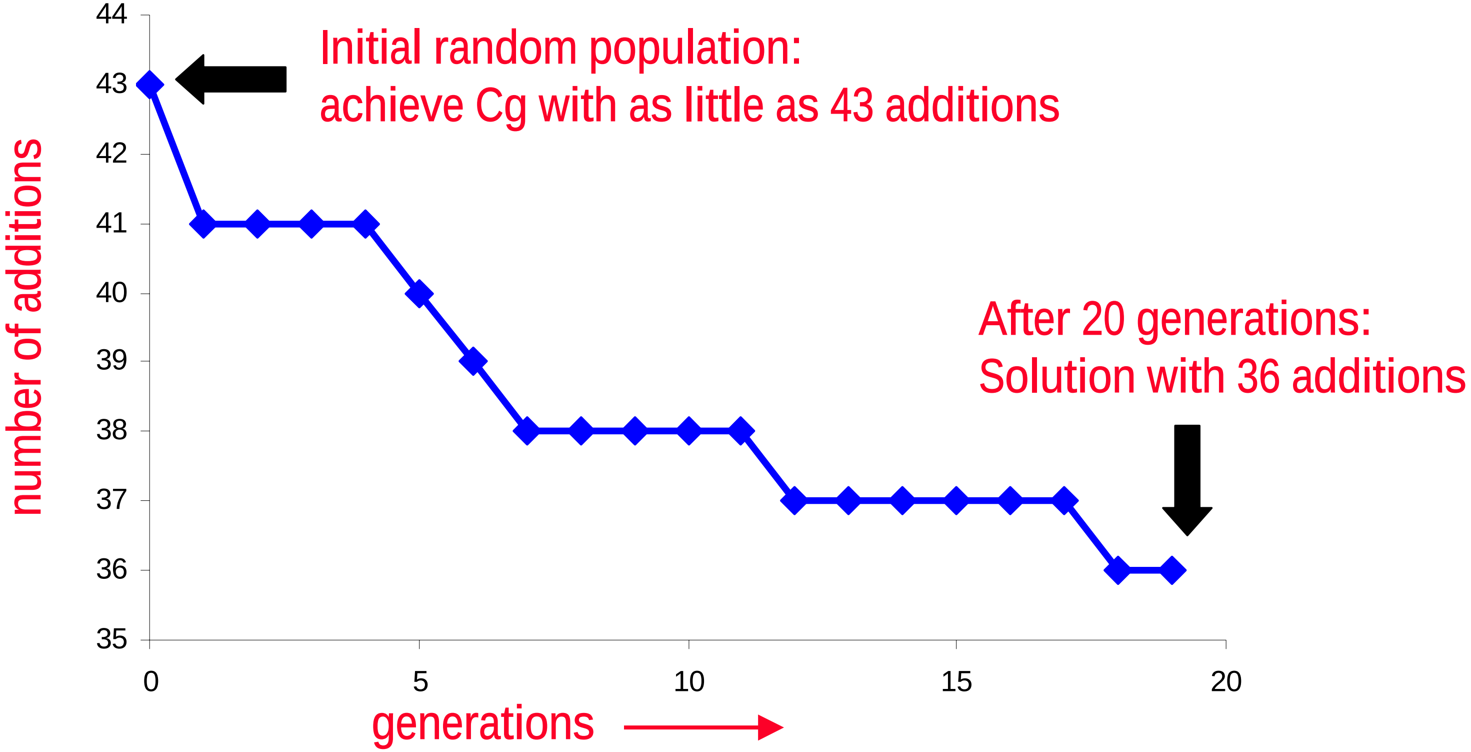
- 3 searches methods were compared
- entire framework implemented as part of SPIRAL (www.spiral.net)

♣*MAD Decoder by Robert Mars, http://www.underbit.com/products/mad*

Example: Evolutionary Search

Evolutionary Search DCT of size 8 with 12 constants

- Q = cg > 8.82, exact DCT has 8.8259
- constant bit length in [0..31]



Initial random population:
achieve Cg with as little as 43 additions

After 20 generations:
Solution with 36 additions

Choosing 31 bits for all constants: 126 additions

Summary of Search Comparison

	Number of Additions (fewer is better)			
	8 pt. DCT-II (8.82 dB cg)	16 pt. DFT (conv. err = 1)	32 pt. DCT-II (LC MP3)	18x36 IMDCT (LC MP3)
initial (31 bits)	126	500	1222	643
global	40	168	*408*	182
evol.	*36*	185	490	212
greedy (top-down)	56	158	417	*170*
greedy (bottom-up)	57	*154*	n/a	n/a

One search method alone is not sufficient — each search performs differently depending on transform and quality measure

Approximation of DCT within JPEG

- Approximate DCT-II inside JPEG while retain images of reasonable quality
  - Q = Peak Signal to Noise Ratio (decibels) of decompressed JPEG image against the original uncompressed input image.

$$\mathrm{PSNR} = 20 \times \log_{10}\left(\frac{255}{\mathrm{RMSE}}\right)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{512 \times 512}\sum_{i}^{512}\sum_{j}^{512}\left[D(i,j) - O(i,j)\right]^2}$$

  - Q Threshold
    - Test Image: Lena, 512x512 pixel, 8-bit grayscale
    - PSNR must be at least 30 decibels or image becomes noticeably lossy).

Approximation of DCT within JPEG

- Before approximating, the original DCT♣ requires 261 additions and produces a Lena image with a PSNR of 37.6462 dB.

Method	# Additions	PSNR
global	37	30.0354
evolutionary	67	36.5323
greedy (t-d)	*28*	*32.4503*

- Compare constants global vs. greedy search:
  - Global: [3/2, 3/2, 3/2, 3/2, 3/2, 3/2, 3/2, 1/2, -1/2, 1,
-1/2, -1/2, 1/2, -1/2, -1, 1, -1, -1/4, 1/2, -1/4]
  - Greedy: [3/2, 1, 1, 1, 1, 1, 1, 1/2, -1/2, 1, -1/2,
0, 1/2, 0, -1, 1, -1, 0, 1/2, -1/4]
  - Greedy succeeds in zeroing 3 constants that affect the high frequency (HF) outputs ‘thrown away’ by JPEG

♣*Base on source from Independent JPEG Group (IJG), http://www.ijg.org*

Summary

- Application specific tuning yields ample opportunities for optimization
- The optimization flow can be automated
  - algorithm selection and manipulation
  - arithmetic reduction through search
  - arbitrary quality measures supported
- Details of the arithmetic reduction is non-trivial
  - non-monotonic relation between Q and C
  - different search methods succeed in different scenarios
- The results of this study needs to be combined with other aspects of DSP domain-specific high-level synthesis